NEURAL NETWORK BASED LEAK LOCALIZATION IN PIPELINES

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

by

Jayesh R Kulkarni

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

September 1996

CERTIFICATE

This is to certify that Jayesh R Kulkarni, Roll no. 9420406 has carried out the thesis work titled "NEURAL NETWORK BASED LEAK LOCALIZATION IN PIPELINES" under our supervision and the same has not been submitted elsewhere for a degree.

Dr. P. R. K. Rao	Dr. R . Sharan
Professor	Professor
Department of Electrical Engineering	Department of Electrical Engineering
I. I. T. Kanpur	I. I. T. Kanpur

September, 1996

ACKNOWLEDGMENT

With no specific order, I would like to say thank you to:

Dr. P. R. K. Rao and Dr. R. Sharan for their constant guidance and constructive criticism during the thesis work.

Richard Bach, Herman Hesse and Henry David Thoreau for reminding me what the purpose of life is.

Sanjudada for introducing me to comic side of life.

Dr. C. P. Gadgil for taking keen interest in all my activities and supporting me all the time.

All my friends for paying least attention to my darker sides, and most to my brighter sides, if any.

Bapukaka for inspiring me by being a living example of dedication and sincerity.

All the good hearted people, who have gone out of the way for helping me, directly or indirectly.

Control Group, Ghat Mandal, Joggers club and Circ-Lib for making the life at IIT Kanpur a memorable one.

Rahul and Shalaka for making my life enjoyable.

Abstract

Pipelines carrying water, oil, petroleum products, sewage etc. form the arteries and veins of modern civilization. Leakage in these pipelines leads to wastage of natural resources, environmental damage, health hazard etc. Locating leaks in pipelines thus becomes an important activity.

In this thesis, Artificial Neural Networks (ANNs) have been used to solve the problem of leak localization in pipelines. Two strategies have been suggested for the solution. The first strategy involves use of Multi-Layer Perceptron (MLP) for classifying correlation patterns of pressure signals associated with different leak locations. This strategy was used for the single leak case.

In the second approach Time Delay Neural Network (TDNN) have been used to extract the temporal as well as the spatial variations of pressure signals. This strategy was used for single leak and double leak cases.

Both MLP and TDNN follow supervised learning. They need large amount of data for their training. Due to non-availability of such data from real life situations, evaluation of the performance of the proposed strategies was carried out with simulated data sets. Performance of MLP was tested on a limited set of real data as well.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Pipelines provide a vital national service with minimum disturbance to the environment. They bring water and fuel to our homes and factories, and carry away wastes. The evergrowing energy needs of the modern world are creating demands for the transportation of oil and gas, thousands of kilometers from remote sources to centers of population, where they can be used.

1.1 Issues In Pipeline Monitoring

Compared to industrial plants, pipelines are in general, not well instrumented. Flowrate sensors and pressure sensors, are mostly available at both ends of pipeline section. A typical pipeline monitoring system involves a SCADA (Supervisory Control And Data Acquisition) system, where the pressure and flow sensors are used as inputs. Leak detection and leak localization are some of the important issues in monitoring of pipelines. Leakage in pipelines (because of corrosion, damage, poor quality/workmanship etc.) is a undesirable condition. Detection and localization of leak is essential for safe, environmental and economic operation.

1.1.1 Some Preliminaries

Leakage in a pipeline carrying liquid/gas can be defined as a situation in which, the material (liquid/gas) passing through the pipeline comes out of it from unwanted location.

Stated in a different way, if M_i is the total mass of the material which is fed into the pipeline, M_o is the total mass of the material coming out from the desired location of pipeline and M_r is the total residual mass in the pipeline, then the condition in which $M_i > M_o + M_r$, leakage is said to be present in the pipeline.

The following cases have to be distinguished with respect to leak detection methods.

- **Medium :** Liquid, gas and multiple phase fluids are the different media possible in pipelines. Accuracy of leak detection generally depends on the medium. Generally leak localization methods for pipelines carrying gases are less accurate.

- **Size of leak :** Detecting smaller leaks, is in general more difficult. Leak localization scheme which is effective for one size of leak may not be useful at all for other size of leaks.

- **Development of leak :** Leaks generated because of corrosion are slowly developed leaks while leaks generated because of damage or accident are abruptly generated leaks. Detecting slowly developed leaks is difficult.

1.1.2 Leak Detection

The traditionally used method [2] for leak detection was mainly based on balancing the input and output flows, by constantly comparing the sum of the delivery quantities to the amount pumped into the pipeline system. Online detection of leaks is possible using this method. Normally Coriolis flow meters are used for this balancing of input and output flows [2].

The main advantage of using Coriolis flow meter over other flow meters is that its output is independent of density variations and hence it gives indication of mass flow and not volume flow. This method however cannot be used for detecting leaks where leak flow is less than 5% of the total mass flow [2].

Another method used for the leak detection is based on pressure changes in the pipeline [1]. For this purpose it is necessary that the pressure profile of the pipeline is known. Any drop in the pressure at both ends of the pipeline section indicates presence of leak in that section. For this method pressure sensors placed at extreme ends of pipeline section are used.

1.1.3 Leak Localization

Liquids escaping through a crack cause a noise, which travels as a pressure wave along the pipe and can be detected using a hydrophone[1]. Two hydrophones, one on either side of the leak, will pick up the same signal but at different times, proportional to the distance each hydrophone is placed from the leak. The signal can be *correlated* statistically by imposing a variable time delay on the signal from one or the other hydrophone. The time delay which gives the peak correlation can be converted into a distance, knowing the velocity of pressure waves through the pipeline, the velocity being function of properties of the liquid and the dimensions of the pipe. Thus the location of leak can be detected. The *leak noise correlator* has been used successfully in locating leaks.

Yet another approach of leak localization involves use of mathematical models of pipelines [14]. Based on the mass balance and momentum balance equations, friction laws and various simplifying assumptions, nonlinear partial differential equations are obtained. The influence of leak flow is modeled in the equations by a leak flow vector. These methods require precise knowledge of the pipeline system. The computational effort becomes too large. These methods usually take hours [14] for locating leaks. The influence of drift effects of the sensors and pipeline need to be suppressed by special methods. Because of all these factors, this method is too complex to be of practical interest.

Correlation method of leak localization is popular a method because of its simplicity in actual implementation. This method however fails in multiple leak cases. This method is also unsuitable for complex pipeline networks.

1.2 Motivation For The Thesis

People in historic periods, used to detect the direction and distance of enemy military from the ground vibrations created by running horses. Though unaware of the laws that govern these ground vibrations, they still could detect the exact location of enemy. This was because of the way human beings learn. Experience, which is supposed to teach human beings a lot, is nothing but ability to generalize the way a system behaves from its cause and effect relationship. Such ability of human brain to learn through experience, not needing rules , is mainly because of the structure of brain.

ANN is an adaptive parallel machine that draws its inspiration from the way the human brain works. ANNs are massively parallel interconnected networks of simple elements (called neurons) intended to interact with the real world in the same way as the biological nervous system. ANNs have been used successfully to solve a wide variety of science and technology problems that involve extracting useful information from complex or uncertain data. A neural network can ideally learn to represent any pattern, its architecture permitting.

Coming to the problem of leak localization, the parametric models obtained from theoretical modeling are too complex to implement and nonparametric models (use of correlation techniques) become unusable for multiple leak cases and for complicated pipeline networks. This leads to a thought of using model independent method of using ANNs for solving the leak localization problem. They have been used in the field of monitoring of pipelines [9] for finding out percentage volumes of different phases in multiphase flow.

1.3 Outline Of The Thesis

For the solution of the leak localization problem using ANNs, certain assumptions have to be made. These assumptions include size of leak, development of leak, number of maximum leaks at a time etc. These assumptions define the exact scope of the problem. These have been discussed in *Chapter 2.* This chapter also gives background material about leak noise, cross correlation and ANNs.

Chapter 3 describes one of the two methods used for leak localization in this thesis. MLP , a commonly used ANN, was used to differentiate between correlation patterns generated because of different leak locations. For this, correlation pattern along with the corresponding location of leak were fed to the MLP with suitable architecture, during the training phase. The MLP trained in this fashion was able to identify certain features of correlation pattern to predict the corresponding leak location.

Chapter 4 discusses other method of solving the same leak localization problem. The ANN used for this method was TDNN. In this method, pressure signals were directly used as the inputs to the TDNN. In the training phase, TDNN was trained to capture the spatial as well as temporal nature of pressure signals. Corresponding to different leak locations, the nature of pressure signal changes. TDNN extracts these changes in pressure signal to predict the leak location. This method was used for single as well as double leak cases.

Chapter 5 concludes the thesis, outlining the significance of the results obtained. It also indicates the scope for future work.

ANNs employed in this work have been described only to the extent required. The focus has been on the problem of leak localization where ANNs have been logically employed as tools. Bibliography at the end of the thesis lists all technical material that was consulted for the development of the thesis.

Chapter 2

Issues In Leak Localization

2.1 Assumptions Made For The Thesis

The pipeline section under consideration is assumed to be perfectly horizontal without any curves and without elevation difference along the pipeline. The pipeline flow is assumed to be one dimensional. The flow in the pipeline, during normal operation, is assumed to be steady state with no temporal variations in the pressure profile across the pipeline. This implies that, during normal operations (i.e. no leak conditions)

$P(x = x_a, t) = constant.$

Thus along the pipeline, the pressure goes on decreasing downstream. Any permanent change in the pressure at any point in the pipeline indicates presence of leak in the pipeline. All sizes (small to large) of abrupt leaks have been considered in the present thesis. The assumption that the leaks are generated abruptly, physically implies that the leaks are generated by impacts or accidents, and not by corrosion etc. This nature of leak generation means that the pressure profile in the pipeline was normal just before the generation of leak.

For the sake of simplicity, the possibility of having more than two leaks at a time in the pipeline is not considered. The thesis aims at solving the leak localization problem using ANNs for the particular type of conditions as mentioned above. The leak monitoring process is continuous.

In liquids, transmission of sound in the form of pressure waves is affected by such factors as temperature, chemical composition of medium. All such properties of the medium are assumed to be constant throughout.

2.2 Generation Of Leak Noise

The concept of leak noise has been used in one of the two approaches developed for leak localization in this thesis. In that approach correlation of leak noise at two different locations in the pipeline is used as the input to the ANN. This section briefly discusses the generation and propagation of leak noise in pipelines.

A localized variable source of mechanical force imposes unbalanced forces on neighboring volume elements. The propagation of the resulting motion-strain effects away from the source, results in a longitudinal compression wave that transmits mechanical, or acoustic energy away from the source.

The propagation of sound in liquids is a mechanical phenomenon and depends on the mechanical properties of the medium [5]. In particular the inertial and elastic properties of an elemental volume are of interest. A net force across the elemental volume element, caused by the sudden exposure of liquid to very low pressure because of leak, results in an acceleration opposed by inertial properties, and mechanical strain is created in the element. The total energy involved in these mechanical effects includes kinetic energy of motion and the stored potential energy represented by internal strain.

Hydrophone constitutes the basic acoustic receiving sensor element[5]. It takes advantage of piezoelectric phenomenon. The piezoelectric effect is normally associated with the behavior of certain crystalline substances. When these materials are subjected to a mechanical stress, electrical charges of opposite polarity appear on portions of the crystal surfaces. The magnitude of these charges is in proportion to the magnitude of the stress, and charges change in polarity as the stress changes from a compression to a tension.

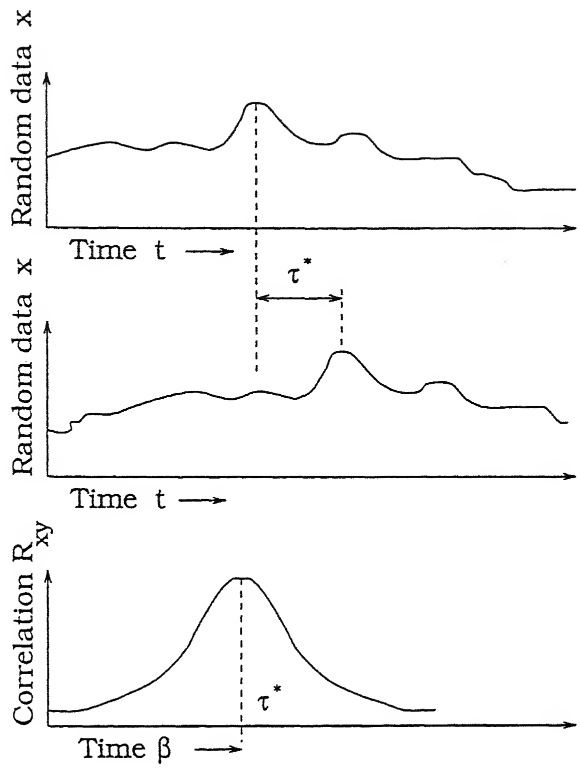

Figure 2.1: Correlation Function.

2.3 Cross Correlation In Measurement

Cross correlation is a method of determining a relationship between two sets of data[4]. Cross correlation finds numerous applications in diverse fields.

The cross correlation function of two sets of random data, such as data from hydrophones, describes the time or phase dependence of one set of data on the other.

The cross correlation of the signals $x(t)$ and $y(t)$ may be obtained from the average of the product of the two values over the observation time T as,

$$R_{xy}(\beta) = lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)y(t+\beta)dt$$

This can be interpreted as the time average of the product of two signals with one of the signal shifted (advanced) in time by time interval β. The result $R_{xy}(\beta)$ is a function of the relative time shift β. Since physically, time delays are realizable and time advancements are not, the following equivalent expression for cross corre-

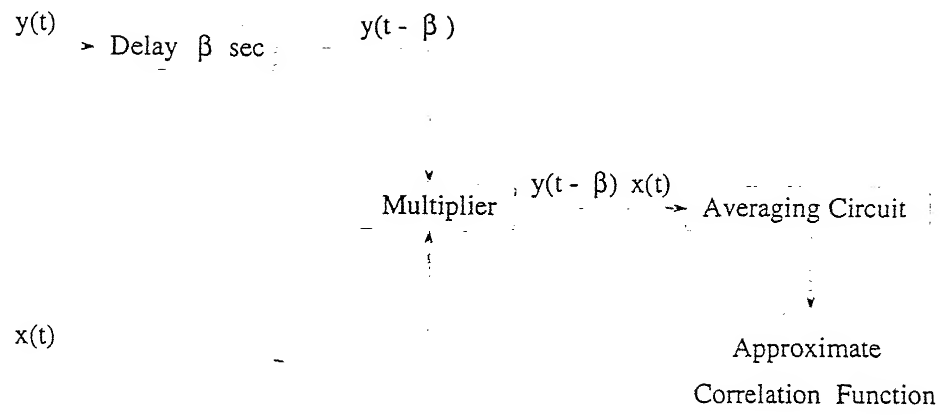

Figure 2.2: Cross Correlator.

lation is used.

$$R_{xy}(\beta) = lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)y(t-\beta)dt$$

Block diagram of *correlator* is shown in Fig. 2.2 .

2.4 Introduction To ANN

ANN models go by many names such as connectionist models [8][10][11], parallel distributed processing models etc. All these models attempt to achieve good performance via dense interconnection of simple computational elements. In this respect ANN structure is based on our present understanding of biological nervous system. Instead of following a program of instructions sequentially as in a computer, neural net models use massively parallel networks composed of many computational elements connected by links with variable weights.

Computational elements or nodes used in neural net models are nonlinear, are typically analog, and may be slow compared to modern digital circuitry. The simplest node, a perceptron sums N weighted inputs and passes the result through a nonlinearity. Three types of nonlinearities are common viz. hard limiter, threshold logic element, and sigmoidal nonlinearity.

Neural net models are specified by the net topology, node characteristics, and

training or learning rules. These rules specify an initial set of weights and indicate how weights should be adapted during training to improve performance.

Since the MLP was used in the thesis, a brief idea about its architecture and the BP algorithm is given here.

2.4.1 Multi-Layer Perceptron

MLP is a feed forward neural network architecture in which a number of perceptrons are arranged in layers with weighted interconnections. Architecture of a generalized MLP with two hidden layers is shown in Fig. 2.3 . Each neuron or perceptron, which is the smallest building block of the network, does a nonlinear transformation of the weighted sum of its inputs. In general, continuous differentiable nonlinear functions are used. Without the nonlinearity the network can be collapsed into a single layer linear transformation[8]. The weights of the network form the Long Term Memory (LTM) and the neuron outputs forms the Short Term Memory (STM). Learning in MLP is supervised. MLP shows following distinct properties.

1. Arbitrary continuous mapping can be uniformly approximated by a feed forward neural network (MLP), with two hidden layers[10].
2. Similar internal representations produce similar outputs.

2.4.2 Back-Propagation Algorithm

The BP algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multilayer feedforward perceptron and the desired output. It requires a continuous differentiable nonlinearity.

The following notations will be used in the further discussion:

N_l total number of neurons in layer l.

w_{ij} weight from neuron j in input layer to neuron i in first hidden layer. All neurons are labeled continuously from 0 to $N_l - 1$.

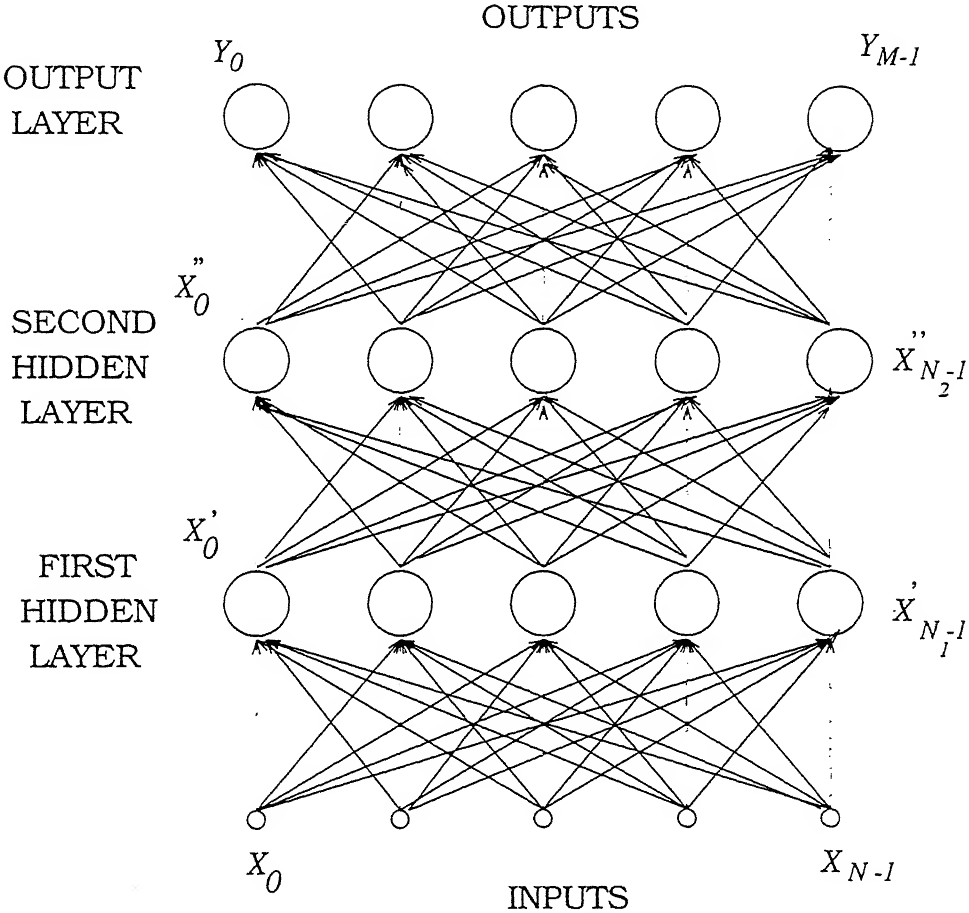

Figure 2.3: Architecture of a generalized MLP with two hidden layers.

w'_{ij} weight from neuron j in first hidden layer to i in second hidden layer.

w''_{ij} weight from neuron j in second hidden layer to i in output layer.

y_i weighted sum of inputs (activation) of the i^{th} neuron.

$s(.)$ the nonlinear sigmoidal function. It is assumed that all nodes have same kind of nonlinearity.

d the desired output of the output neuron.

x'_j the output of neuron j in first layer.

x''_k the output of neuron k in second layer.

θ'_k internal offset in neuron k of the first hidden layer.

θ''_l internal offset in neuron l of the second hidden layer.

The following discussion assumes a sigmoidal nonlinearity. The iterative steps of the BP algorithm are as follows:

- **Step 1: Initialize Weights and Offsets**

  Set all weights and neuron offsets to small random values.

- **Step 2: Present Input and Desired Outputs**

  Present an input vector $x_0, x_1, ...x_{N-1}$ and specify the desired outputs d_0, $d_1, ...d_{M-1}$.

- **Step 3: Calculate Actual Outputs**

  Use the sigmoidal nonlinearity and use following formulaes to calculate outputs $y_0, y_1, ...y_{M-1}$.

$$x'_j = f(\sum_{i=0}^{N-1} w_{ij}x_i - \theta_j) \quad 0 \le j \le N_1 - 1$$

$$x''_k = f(\sum_{j=0}^{N_1-1} w'_{jk}x'_j - \theta'_k) \quad 0 \le k \le N_2 - 1$$

$$y_l = f(\sum_{k=0}^{N_2-1} w''_{kl}x''_k - \theta''_l) \quad 0 \leq l \leq M-1$$

- **Step 4: Adapt Weights**

  Use a recursive algorithm starting at the output neurons and working back to the first hidden layer. Adjust weights by

  $$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x'_i$$

  In this equation $w_{ij}(t)$ is the weight from hidden neuron j at the instant t, x'_i is either the output of neuron i or is an input, η is a gain term, and δ_j is an error term for neuron j. If the neuron j is an output neuron, then

  $$\delta_j = y_j(1-y_j)(d_j - y_j)$$

  where d_j is the desired output of the node j and y_j is the actual output.

  If the neuron j is an internal hidden neuron, then

  $$\delta_j = x'_j(1-x'_j) \sum_k \delta_k w_{jk}$$

  where k is over all neurons in the layers above neuron j. Internal node thresholds are adapted in a similar manner by assuming that, they are connection weights on links from auxiliary constant-valued inputs. Convergence is sometimes faster if a momentum term is added and weight changes are smoothed by

  $$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x'_i + \alpha(w_{ij}(t) - w_{ij}(t-1))$$

  , where $0 \leq \alpha \leq 1$.

- **Step 5: Repeat by Going to Step 2**

2.5 Implementation

A program simulating generalized architecture of MLP was written in C++ language which was run on HP-9000 super-mini computer. This program has following features.

The MLP can have maximum 50 number of inputs. The maximum number of outputs possible is 5. The MLP can have maximum 3 hidden layers with each layer consisting of maximum 25 neurons. All these parameters which specify the architecture of MLP are user programmable. The program when run successfully, generates a weight file which contains all the weights of the MLP. The program generates error file which contains error found out during each iteration. The weights are selected randomly at the start of the BP algorithm. The program has an option to feed the weights through a file so that the user can decide the starting weights.

Chapter 3

MLP As A Correlation Classifier

This chapter discusses the strategy of using the MLP to distinguish between the correlation patterns associated with different leak locations. This method was used for the single leak case only.

3.1 Introduction

Figure 3.1 shows a pipeline section of length D. Sensors A and B are the two hydrophones located at the extreme ends of the pipeline section. In the case when there is no leak,

$P(x = 0, t) = a_1$; a constant

$P(x = D, t) = a_2$; a constant.

Now if there is presence of leak L at a distance d, from sensor A, then because of acoustic waves, the pressure at both sensors A and B will vary.

Since in this thesis, only the abrupt leaks are considered, the pressures at sensors A and B, just before the leak is developed will be a_1 and a_2 respectively. Because of the acoustic waves, a transient is observed in pressures at both the sensors A and B. After sufficient time, which depends on pipeline dimensions and various other parameters, the pressures attain steady state values c_1 and c_2 at sensors A and B respectively. The transients observed are solely because of the leak. The time the transient takes to reach sensors A and B depends on the location of the leak. The

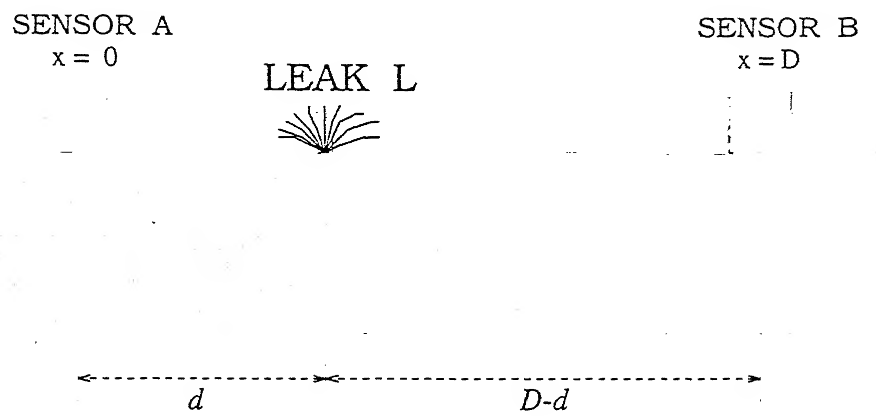

Figure 3.1: Pipeline Section.

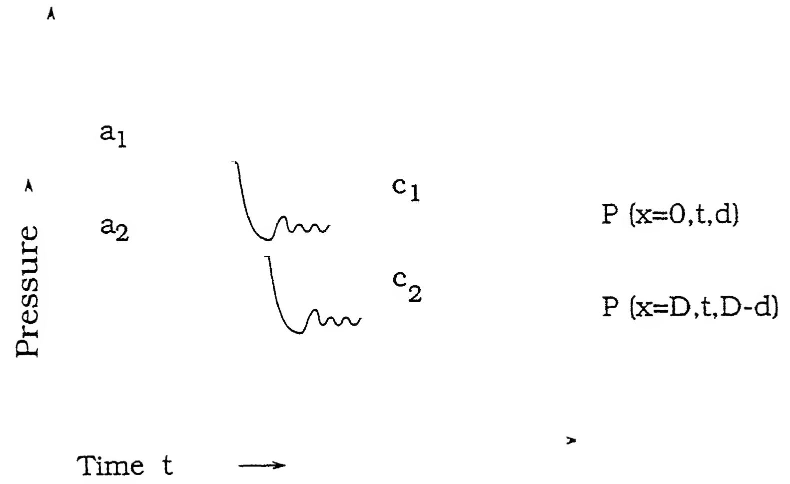

Figure 3.2: Pressures at A and B.

correlation of the transient signals $P(x = 0, t, d)$ and $P(x = D, t, D - d)$ gives a characteristic pattern for a particular location of leak.

3.2 Strategy For Leak Localization

Let $P_A \; = \; P(x = 0, t, d)$

and $P_B \; = \; P(x = D, t, D - d)$

In the condition when there is no leak, the cross correlation function will be,

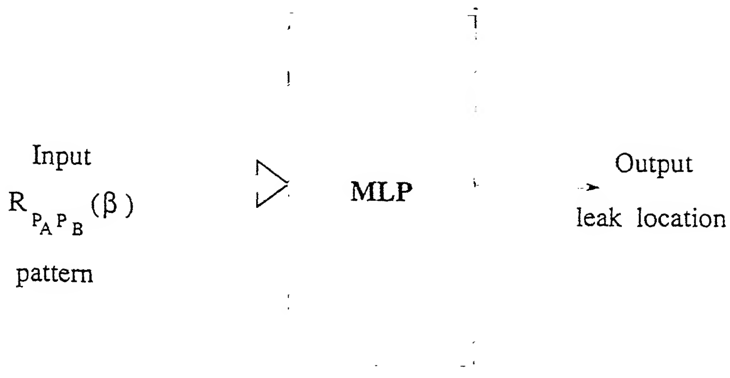

Figure 3.3: Strategy Of Leak Localization Using MLP.

$$R_{P_AP_B}(\beta) \;=\; lim_{T\to\infty} \frac{1}{T}\int_0^T P(x=0,t)P(x=D,t-\beta)dt$$

$$= a_1a_2$$

When a leak L is generated, during the transients the cross correlation function will be,

$$R_{P_AP_B}(\beta) \;=\; lim_{T\to\infty} \frac{1}{T}\int_0^T P(x=0,t,d)P(x=D,t-\beta,D-d)dt$$

And after the transients have died down the cross correlation function will be,

$$R_{P_AP_B}(\beta) \;=\; lim_{T\to\infty} \frac{1}{T}\int_0^T P(x=0,t,d)P(x=D,t-\beta,D-d)dt$$

$$= c_1c_2$$

$R_{P_AP_B}(\beta)$ pattern consists of $R_{P_AP_B}$ values for different values of β. A particular $R_{P_AP_B}(\beta)$ pattern corresponds to a particular leak location. Different $R_{P_AP_B}(\beta)$ patterns corresponding to different leak locations are fed to the MLP (with 2 hidden layers) as inputs. The leak location is the output of the MLP. During the training phase, training patterns consisting of leak location and corresponding $R_{P_AP_B}(\beta)$ function are fed to the MLP as output and input respectively.

The MLP is trained to differentiate between the correlation patterns generated because of different leak locations. But since the real life data i.e. correlation patterns are not available to the extent required for training of the MLP, realistic data sets should be generated for the training of the MLP.

3.2.1 Data Set Generation

Choice of functions P_A and P_B is an important decision. Four different functions were used in this thesis. They are as follows

1. **Signal 1:**

$$P(x=0,t,d) \;=\; sin\omega_1(t-d/v) \;+\; c_1$$

$$P(x=D,t,D-d) \;=\; sin\omega_1(t-\tfrac{D-d}{v}) \;+\; c_2$$

2. **Signal 2:**

$$P(x=0,t,d) \;=\; sin\omega_1(t-d/v) \;+\; sin\omega_2(t-d/v) \;+\; sin\omega_3(t-d/v) \;+\; c_1$$

$$P(x=D,t,D-d) \;=\; sin\omega_1(t-\tfrac{D-d}{v}) + sin\omega_2(t-\tfrac{D-d}{v}) + sin\omega_3(t-\tfrac{D-d}{v}) + c_2$$

3. **Signal 3:**

$$P(x=0,t,d) \;=\; e^{-\alpha_1 t}\, sin\omega_1(t-d/v) \;+\; c_1$$

$$P(x=D,t,D-d) \;=\; e^{-\alpha_1 t}\, sin\omega_1(t-\tfrac{D-d}{v}) \;+\; c_2$$

4. **Signal 4:**

$$P(x=0,t,d) \;=\; e^{-\alpha_1 t}\, sin\omega_1(t-d/v) \;+\; e^{-\alpha_2 t}\, sin\omega_2(t-d/v) \;+\; c_1$$

$$P(x=D,t,D-d) \;=\; e^{-\alpha_1 t}\, sin\omega_1(t-\tfrac{D-d}{v}) \;+\; e^{-\alpha_2 t}\, sin\omega_2(t-\tfrac{D-d}{v}) \;+\; c_2$$

The corresponding cross correlation functions are as follows.

1. **For Signal 1:**

$$R_{P_AP_B}(\beta) \;=\; lim_{T\to\infty}\frac{1}{T}\int_0^T P(x=0,t,d)P(x=D,t-\beta,D-d)dt$$

$$=\; \tfrac{1}{2}(\; cos\omega_1(\tfrac{D-2d}{v}-\beta)) \;+\; c_1c_2$$

2. **For Signal 2:**

$$R_{P_AP_B}(\beta) \;=\; lim_{T\to\infty}\frac{1}{T}\int_0^T P(x=0,t,d)P(x=D,t-\beta,D-d)dt$$

$$=\; \tfrac{1}{2}(\; cos\omega_1(\tfrac{D-2d}{v}-\beta) \;+\; cos\omega_2(\tfrac{D-2d}{v}-\beta) \;+\; cos\omega_3(\tfrac{D-2d}{v}-\beta)) \;+\; c_1c_2$$

3. **For Signal 3:**

$$R_{P_A P_B}(\beta) \;=\; lim_{T\to\infty} \frac{1}{T}\int_0^T P(x=0,t,d)P(x=D,t-\beta,D-d)dt$$

$$= \tfrac{1}{2}(e^{-\alpha_1\beta}\,cos\omega_1(\tfrac{D-2d}{v}-\beta)) \;+\; c_1c_2$$

4. **For Signal 4:**

$$R_{P_A P_B}(\beta) \;=\; lim_{T\to\infty} \frac{1}{T}\int_0^T P(x=0,t,d)P(x=D,t-\beta,D-d)dt$$

$$= \tfrac{1}{2}(e^{-\alpha_1\beta}\,cos\omega_1(\tfrac{D-2d}{v}-\beta) \;+\; e^{-\alpha_2\beta}\,cos\omega_2(\tfrac{D-2d}{v}-\beta)) \;+\; c_1c_2$$

3.2.2 Implementation

Programs were written in C language which were run on HP-9000 super-mini computer, to generate data sets. Data generated by these programs have been used in all the experiments carried out with this method as part of the thesis. When fed with necessary parameter values, the programs generate output files, which contain the data sets. Each data set generated, contains 101 number of $R_{P_A P_B}(\beta)$ patterns corresponding to different leak locations. Each data in turn contains 21 sampled values of $R_{P_A P_B}(\beta)$ pattern for different values of β.

3.3 Structure Of The MLP

The MLP was used for classifying different correlation patterns corresponding to different leak locations.

A particular location of leak corresponds to a characteristic correlation pattern. And the MLP was trained to identify the correct correlation pattern which in turn leads to localization of the leak.

The MLP used for this method had twentyone inputs (corresponding to $R_{P_A P_B}(\beta)$ pattern). Since only single case was assumed in this method, the MLP had only one output (corresponding to the leak location). The MLP had two hidden layers with fifteen neurons in the first hidden layer and nine neurons in the second hidden

ANN	ω_1	c_1	c_2
MLP 1	180	10	9
MLP 2	430	9	8
MLP 3	200	8.5	7
MLP 4	300	11	9.5

Table 3.1: Values of parameters for Signal 1 type of models.

layer. The nonlinearity used in each neuron was sigmoidal. The BP algorithm was used for training of the network.

3.4 Training Of The MLP

Data sets generated as discussed in the Section 3.2 were used during the training phase. Out of the 101 patterns generated, 51 patterns were randomly selected for training of the MLP and remaining were used for testing of the MLP. At the start of the training phase all the weights were randomly generated.

Percentage error e_n for a pattern n was defined as

$$e_n = \left|\frac{expected\ output - actual\ output}{expected\ output} \times 100\right|$$

Percentage error per pattern E was defined as

$$E = (\sqrt{e_1^2 + e_2^2 + \ldots + e_n^2})/n$$

where n is the total number of patterns.

Training of the MLP was stopped when E was less than 2%. Using the above strategy sixteen different MLPs were trained, using all the four types of signals as discussed in Section 3.2 with different values of parameters as given in Tables 3.1 to 3.4 .

ANN	ω_1	ω_2	ω_3	c_1	c_2
MLP 5	180	450	310	10	8
MLP 6	220	430	510	9	7
MLP 7	200	440	315	10	8.5
MLP 8	300	180	140	11	8

Table 3.2: Values of parameters for Signal 2 type of models.

ANN	α_1	ω_1	c_1	c_2
MLP 9	.145	180	10	8
MLP 10	.11	430	9	7
MLP 11	.10	200	10	8.5
MLP 12	.095	300	11	8

Table 3.3: Values of parameters for Signal 3 type of models.

ANN	α_1	α_2	ω_1	ω_2	c_1	c_2
MLP 13	.09	.145	180	450	10	8
MLP 14	.11	.130	220	430	9	7
MLP 15	.10	.140	200	440	10	8.5
MLP 16	.095	.170	300	180	11	8

Table 3.4: Values of parameters for Signal 4 type of models.

3.5 Testing Of The MLP

In the testing phase of the MLP, fifty patterns were fed as the inputs to the MLP. Since the expected output is known, the comparison of this expected output with the output of the MLP (which is indication of leak location) gives the idea about the performance of the MLP. All the sixteen MLPs (which were trained) were tested in the above fashion. Error Ep for a pattern was defined as

$$Ep = \left|\frac{Leak\ Location - Output\ Of\ MLP}{Leak\ Location}\right| \times 100$$

3.5.1 Results

Tables 3.5 to 3.20 give the leak locations, output of MLP, absolute error (i.e. modulus of difference between leak location and output of MLP) and Ep for 25 sample testing patterns for all the 16 MLPs. Figures 3.4, 3.6, 3.8 and 3.10 give the plot of error Ep in leak localization for different leak locations for all the MLPs. Figures 3.5, 3.7, 3.9 and 3.11 give the plot of leak location versus output of MLP for all the MLPs.

3.5.2 Conclusion

The Figures 3.4 to 3.11 show that the MLPs, trained with the randomly chosen 51 patterns were able to predict the output for the remaining 50 patterns with acceptable accuracy. For the total range of leak locations (i.e. from 0 to 1) the percentage error Ep remains mostly within 10%. For the initial leak values from 0 to .2 the percentage error Ep is high because of the fact that the absolute leak values are very small. The absolute error as shown in corresponding tables indicate that the absolute error for these values is not very large.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.069807	0.019807	39.613998
0.100000	0.089539	0.010461	10.461002
0.150000	0.125335	0.024665	16.443342
0.200000	0.178572	0.021428	10.714002
0.250000	0.242312	0.007688	3.075200
0.300000	0.305796	0.005796	1.931995
0.350000	0.362880	0.012880	3.679999
0.400000	0.413058	0.013058	3.264502
0.450000	0.458454	0.008454	1.878672
0.500000	0.501682	0.001682	0.336397
0.550000	0.545016	0.004984	0.906186
0.600000	0.590306	0.009694	1.615673
0.650000	0.639404	0.010596	1.630150
0.700000	0.694511	0.005489	0.784142
0.750000	0.756532	0.006532	0.870935
0.800000	0.820379	0.020379	2.547376
0.850000	0.874256	0.024256	2.853646
0.900000	0.910034	0.010034	1.114892
0.950000	0.929212	0.020788	2.188212
1.000000	0.936682	0.063318	6.331801

Table 3.5: Sample Results Of Testing For MLP 1.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.058576	0.008576	17.151997
0.100000	0.082157	0.017843	17.843000
0.150000	0.129066	0.020934	13.956000
0.200000	0.194082	0.005918	2.958998
0.250000	0.259038	0.009038	3.615201
0.300000	0.312481	0.012481	4.160325
0.350000	0.355500	0.005500	1.571434
0.400000	0.395332	0.004668	1.166999
0.450000	0.441480	0.008520	1.893328
0.500000	0.499392	0.000608	0.121599
0.550000	0.560104	0.010104	1.837091
0.600000	0.609150	0.009150	1.524995
0.650000	0.647411	0.002589	0.398306
0.700000	0.686154	0.013846	1.977997
0.750000	0.736254	0.013746	1.832803
0.800000	0.802147	0.002147	0.268370
0.850000	0.871015	0.021015	2.472352
0.900000	0.920042	0.020042	2.226889
0.950000	0.943726	0.006274	0.660419
1.000000	0.948608	0.051392	5.139202

Table 3.6: Sample Results Of Testing For MLP 2.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.066315	0.016315	32.630005
0.100000	0.086532	0.013468	13.468004
0.150000	0.128902	0.021098	14.065334
0.200000	0.186756	0.013244	6.622002
0.250000	0.246429	0.003571	1.428401
0.300000	0.302588	0.002588	0.862658
0.350000	0.355662	0.005662	1.617713
0.400000	0.406533	0.006533	1.633249
0.450000	0.455296	0.005296	1.176894
0.500000	0.501774	0.001774	0.354803
0.550000	0.546594	0.003406	0.619271
0.600000	0.592135	0.007865	1.310835
0.650000	0.641572	0.008428	1.296612
0.700000	0.695777	0.004223	0.603284
0.750000	0.752563	0.002563	0.341733
0.800000	0.810573	0.010573	1.321621
0.850000	0.868940	0.018940	2.228232
0.900000	0.916797	0.016797	1.866334
0.950000	0.940036	0.009964	1.048841
1.000000	0.941188	0.058812	5.881202

Table 3.7: Sample Results Of Testing For MLP 3.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.066595	0.016595	33.190002
0.100000	0.073634	0.026366	26.366003
0.150000	0.116321	0.033679	22.452671
0.200000	0.189329	0.010671	5.335502
0.250000	0.266263	0.016263	6.505203
0.300000	0.322944	0.022944	7.647991
0.350000	0.358034	0.008034	2.295434
0.400000	0.387255	0.012745	3.186248
0.450000	0.438061	0.011939	2.653109
0.500000	0.527674	0.027674	5.534804
0.550000	0.581453	0.031453	5.718729
0.600000	0.595199	0.004801	0.800172
0.650000	0.616707	0.033293	5.121993
0.700000	0.667027	0.032973	4.710427
0.750000	0.748665	0.001335	0.178003
0.800000	0.835483	0.035483	4.435375
0.850000	0.895190	0.045190	5.316468
0.900000	0.923489	0.023489	2.609889
0.950000	0.925322	0.024678	2.597683
1.000000	0.812264	0.187736	18.773598

Table 3.8: Sample Results Of Testing For MLP 4.

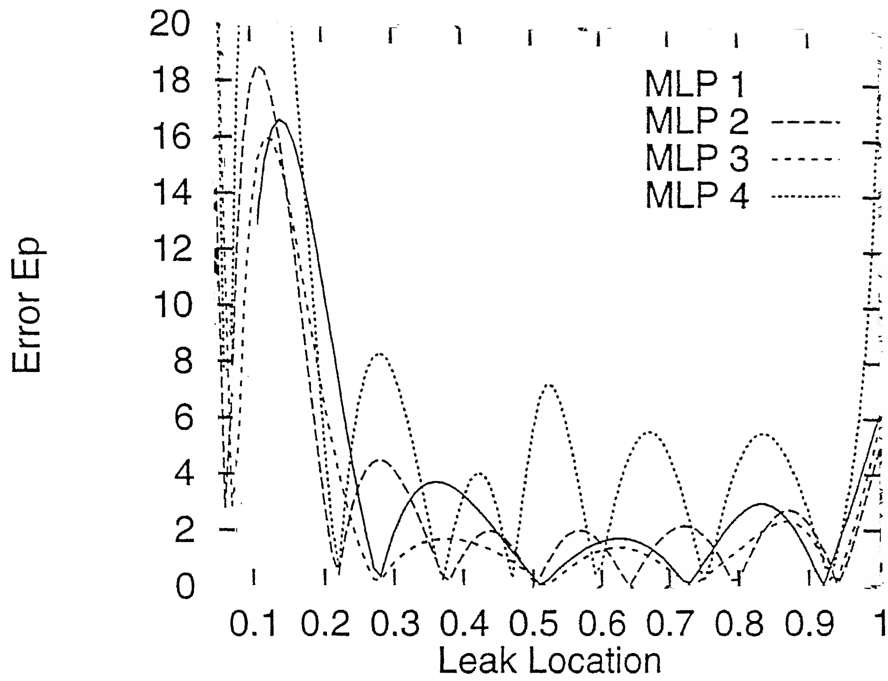

Figure 3.4: Plot Of Error (Ep) In Leak Localization For Different Leak Locations.

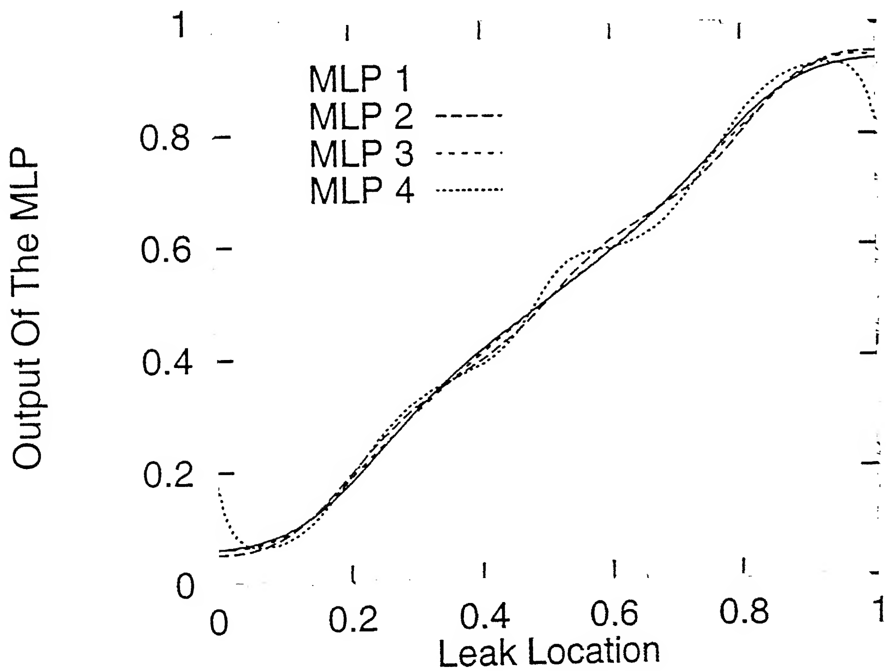

Figure 3.5: Results Of Testing For MLP 1, MLP 2, MLP 3 and MLP 4.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.048755	0.001245	2.489999
0.100000	0.079556	0.020444	20.443998
0.150000	0.142980	0.007020	4.680007
0.200000	0.209140	0.009140	4.570000
0.250000	0.260846	0.010846	4.338396
0.300000	0.302598	0.002598	0.865996
0.350000	0.345140	0.004860	1.388567
0.400000	0.395101	0.004899	1.224749
0.450000	0.452709	0.002709	0.602000
0.500000	0.508505	0.008505	1.700997
0.550000	0.547757	0.002243	0.407815
0.600000	0.593149	0.006851	1.141836
0.650000	0.649666	0.000334	0.051379
0.700000	0.700940	0.000940	0.134289
0.750000	0.749565	0.000435	0.057999
0.800000	0.803147	0.003147	0.393376
0.850000	0.864186	0.014186	1.668937
0.900000	0.918462	0.018462	2.051334
0.950000	0.947985	0.002015	0.212105
1.000000	0.955415	0.044585	4.458499

Table 3.9: Sample Results Of Testing For MLP 5.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.054066	0.004066	8.131995
0.100000	0.082034	0.017966	17.966002
0.150000	0.140874	0.009126	6.084005
0.200000	0.201589	0.001589	0.794500
0.250000	0.255556	0.005556	2.222395
0.300000	0.301677	0.001677	0.558992
0.350000	0.340949	0.009051	2.585999
0.400000	0.406858	0.006858	1.714498
0.450000	0.459555	0.009555	2.123336
0.500000	0.497888	0.002112	0.422400
0.550000	0.552463	0.002463	0.447815
0.600000	0.604102	0.004102	0.683665
0.650000	0.646552	0.003448	0.530454
0.700000	0.689930	0.010070	1.438567
0.750000	0.741521	0.008479	1.130533
0.800000	0.811595	0.011595	1.449376
0.850000	0.868993	0.018993	2.234466
0.900000	0.914028	0.014028	1.558668
0.950000	0.939457	0.010543	1.109788
1.000000	0.945023	0.054977	5.497700

Table 3.10: Sample Results Of Testing For MLP 6.

3.6 Handling Of Real Data

Considering the nature of the problem it is difficult to obtain sufficient amount of data for the training of the MLP. The following strategy proves to be useful for the handling real life situations. For the training of the MLP, use realistically simulated data. To ensure that realistically simulated data represents the real data, the parameters of the model used for simulating data should be chosen such that the error as defined below is minimum.

$\mathcal{E} = \sum_{\beta=-10}^{10} (R_{P_A P_B emp}(\beta) - R_{P_A P_B model}(\beta))^2$

where $R_{P_A P_B emp}(\beta)$ is the empirically obtained data, and $R_{P_A P_B model}(\beta)$ is simulated data. Finally test the MLP with the real life data.

3.6.1 Optimization Of The Error Function $\mathcal{E}$

Steepest descent method[13] was used for the minimization of error function $\mathcal{E}$. All the four types of models were used for the optimization. The model which gives best fit i.e minimum $\mathcal{E}$ was chosen for the data set generation.

Let Y_i represent parameter vector of the model, i.e. for Signal 4 type of model,

$$Y_i = \begin{pmatrix} \alpha_{1i} \\ \alpha_{2i} \\ \omega_{1i} \\ \omega_{2i} \\ c_{1i} c_{2i} \end{pmatrix}$$

where i is no. of iterations.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.052071	0.002071	4.142001
0.100000	0.082017	0.017983	17.983004
0.150000	0.139194	0.010806	7.204006
0.200000	0.204967	0.004967	2.483502
0.250000	0.259340	0.009340	3.735995
0.300000	0.301270	0.001270	0.423332
0.350000	0.343167	0.006833	1.952282
0.400000	0.397022	0.002978	0.744499
0.450000	0.461080	0.011080	2.462228
0.500000	0.505825	0.005825	1.164997
0.550000	0.539385	0.010615	1.929998
0.600000	0.596629	0.003371	0.561833
0.650000	0.656366	0.006366	0.979387
0.700000	0.704699	0.004699	0.671285
0.750000	0.744811	0.005189	0.691867
0.800000	0.789630	0.010370	1.296252
0.850000	0.851946	0.001946	0.228938
0.900000	0.917484	0.017484	1.942668
0.950000	0.953604	0.003604	0.379368
1.000000	0.962477	0.037523	3.752297

Table 3.11: Sample Results Of Testing For MLP 7.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.064019	0.014019	28.038002
0.100000	0.083306	0.016694	16.694002
0.150000	0.126321	0.023679	15.786001
0.200000	0.190949	0.009051	4.525505
0.250000	0.257056	0.007056	2.822399
0.300000	0.310580	0.010580	3.526658
0.350000	0.352293	0.002293	0.655149
0.400000	0.392925	0.007075	1.768753
0.450000	0.442937	0.007063	1.569556
0.500000	0.499632	0.000368	0.073600
0.550000	0.554064	0.004064	0.738903
0.600000	0.604599	0.004599	0.766496
0.650000	0.653029	0.003029	0.466007
0.700000	0.698464	0.001536	0.219430
0.750000	0.743684	0.006316	0.842134
0.800000	0.799331	0.000669	0.083625
0.850000	0.866774	0.016774	1.973412
0.900000	0.919135	0.019135	2.126111
0.950000	0.941742	0.008258	0.869262
1.000000	0.945473	0.054527	5.452699

Table 3.12: Sample Results Of Testing For MLP 8.

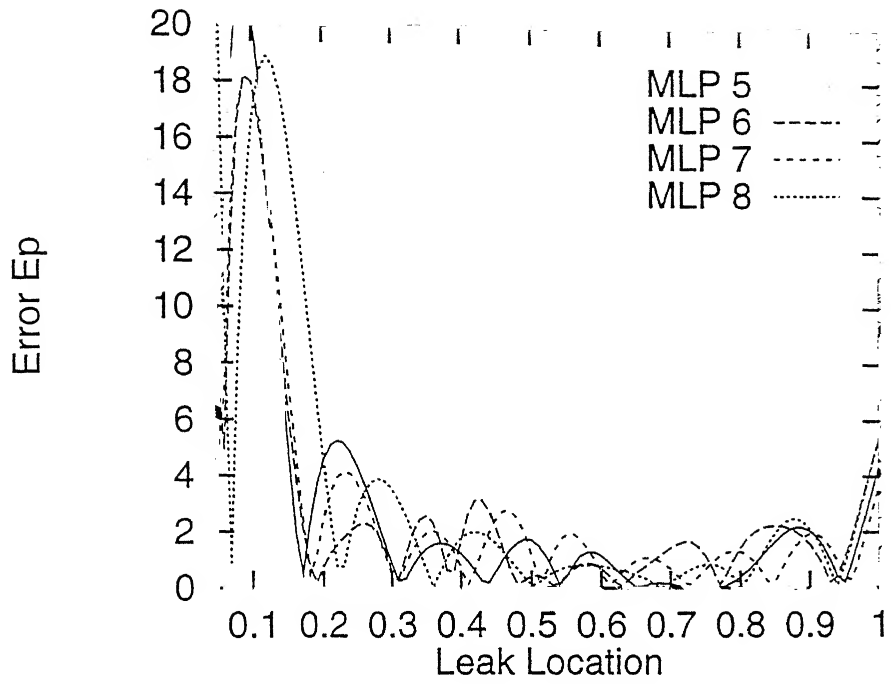

Figure 3.6: Plot Of Error (Ep) In Leak Localization For Different Leak Locations.

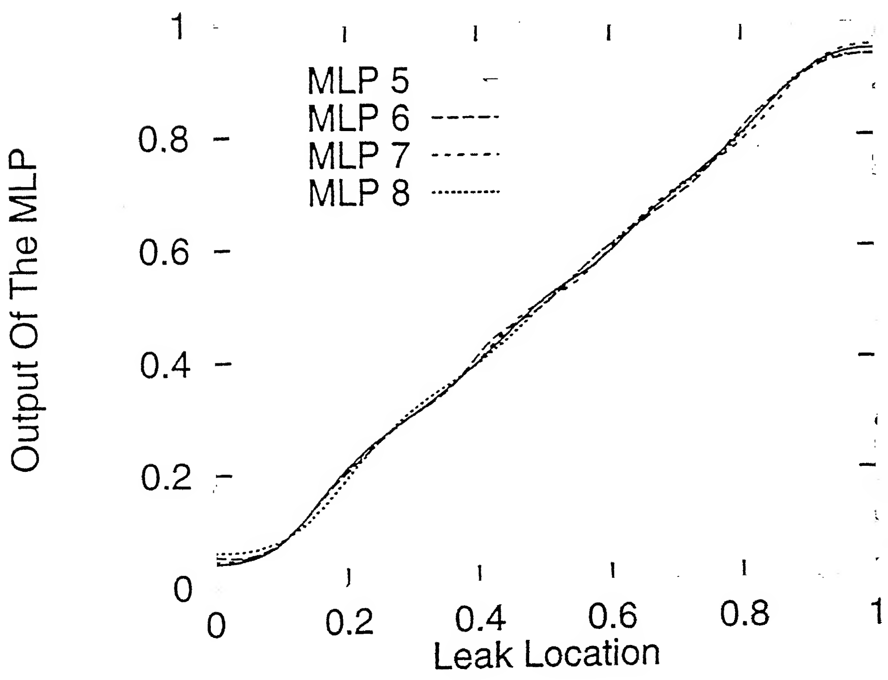

Figure 3.7: Results Of Testing For MLP 5, MLP 6, MLP 7 and MLP 8.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.065891	0.015891	31.781994
0.100000	0.088794	0.011206	11.206001
0.150000	0.128207	0.021793	14.528671
0.200000	0.182378	0.017622	8.811005
0.250000	0.243365	0.006635	2.653998
0.300000	0.303613	0.003613	1.204332
0.350000	0.359905	0.009905	2.830003
0.400000	0.411752	0.011752	2.937995
0.450000	0.459408	0.009408	2.090666
0.500000	0.503567	0.003567	0.713396
0.550000	0.545823	0.004177	0.759461
0.600000	0.588911	0.011089	1.848171
0.650000	0.636360	0.013640	2.098459
0.700000	0.691290	0.008710	1.244281
0.750000	0.753872	0.003872	0.516264
0.800000	0.818155	0.018155	2.269372
0.850000	0.873295	0.023295	2.740587
0.900000	0.911843	0.011843	1.315892
0.950000	0.934245	0.015755	1.658421
1.000000	0.944560	0.055440	5.544001

Table 3.13: Sample Results Of Testing For MLP 9.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.058756	0.008756	17.512001
0.100000	0.077660	0.022340	22.340000
0.150000	0.127307	0.022693	15.128672
0.200000	0.202547	0.002547	1.273498
0.250000	0.268073	0.018073	7.229197
0.300000	0.313891	0.013891	4.630327
0.350000	0.352678	0.002678	0.765145
0.400000	0.394812	0.005188	1.297004
0.450000	0.443577	0.006423	1.427333
0.500000	0.496218	0.003782	0.756401
0.550000	0.548076	0.001924	0.349825
0.600000	0.597030	0.002970	0.495007
0.650000	0.644090	0.005910	0.909228
0.700000	0.692256	0.007744	1.106288
0.750000	0.746191	0.003809	0.507863
0.800000	0.809240	0.009240	1.154996
0.850000	0.873431	0.023431	2.756588
0.900000	0.919442	0.019442	2.160225
0.950000	0.940673	0.009327	0.981789
1.000000	0.942868	0.057132	5.713201

Table 3.14: Sample Results Of Testing For MLP 10.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.052664	0.002664	5.328000
0.100000	0.084097	0.015903	15.903004
0.150000	0.138920	0.011080	7.386674
0.200000	0.201083	0.001083	0.541501
0.250000	0.255819	0.005819	2.327597
0.300000	0.303490	0.003490	1.163334
0.350000	0.349650	0.000350	0.100000
0.400000	0.398030	0.001970	0.492498
0.450000	0.448880	0.001120	0.248889
0.500000	0.500249	0.000249	0.049806
0.550000	0.550143	0.000143	0.025998
0.600000	0.597655	0.002345	0.390838
0.650000	0.643370	0.006630	1.020001
0.700000	0.689683	0.010317	1.473853
0.750000	0.740735	0.009265	1.235334
0.800000	0.800828	0.000828	0.103496
0.850000	0.867116	0.017116	2.013641
0.900000	0.920954	0.020954	2.328224
0.950000	0.948800	0.001200	0.126312
1.000000	0.958093	0.041907	4.190701

Table 3.15: Sample Results Of tseting For MLP 11.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.047514	0.002486	4.972003
0.100000	0.077663	0.022337	22.337004
0.150000	0.141894	0.008106	5.404005
0.200000	0.206737	0.006737	3.368497
0.250000	0.257751	0.007751	3.100395
0.300000	0.301712	0.001712	0.570665
0.350000	0.351105	0.001105	0.315717
0.400000	0.404272	0.004272	1.067996
0.450000	0.437907	0.012093	2.687329
0.500000	0.495082	0.004918	0.983602
0.550000	0.565867	0.015867	2.884908
0.600000	0.605979	0.005979	0.996500
0.650000	0.648732	0.001268	0.195072
0.700000	0.695645	0.004355	0.622145
0.750000	0.742880	0.007120	0.949335
0.800000	0.794061	0.005939	0.742376
0.850000	0.858463	0.008463	0.995643
0.900000	0.925291	0.025291	2.810114
0.950000	0.954660	0.004660	0.490527
1.000000	0.876466	0.123534	12.353402

Table 3.16: Sample Results Of Testing For MLP 12.

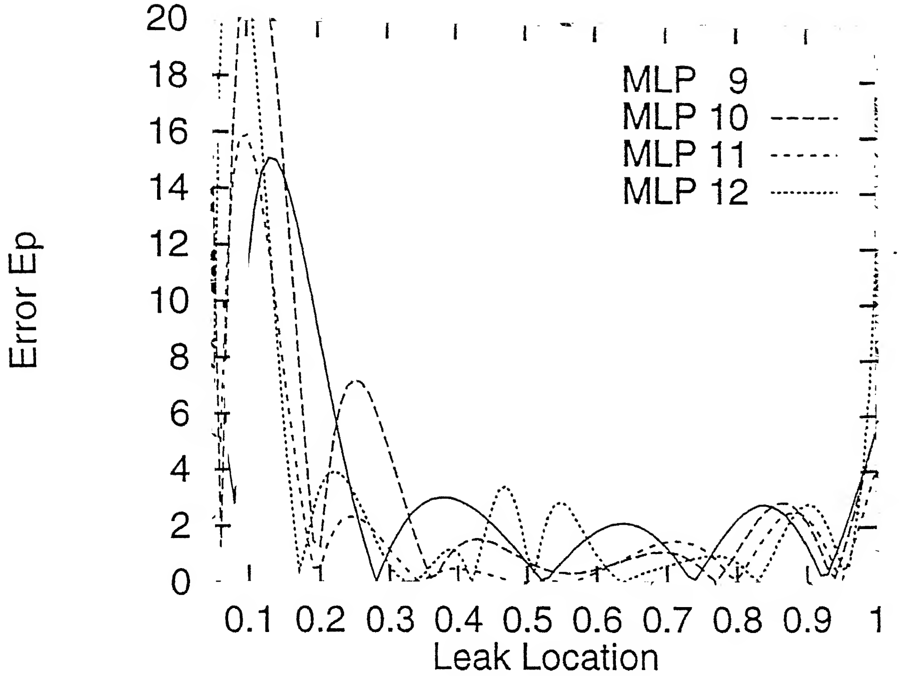

Figure 3.8: Plot Of Error (Ep) In Leak Localization For Different Leak Locations.

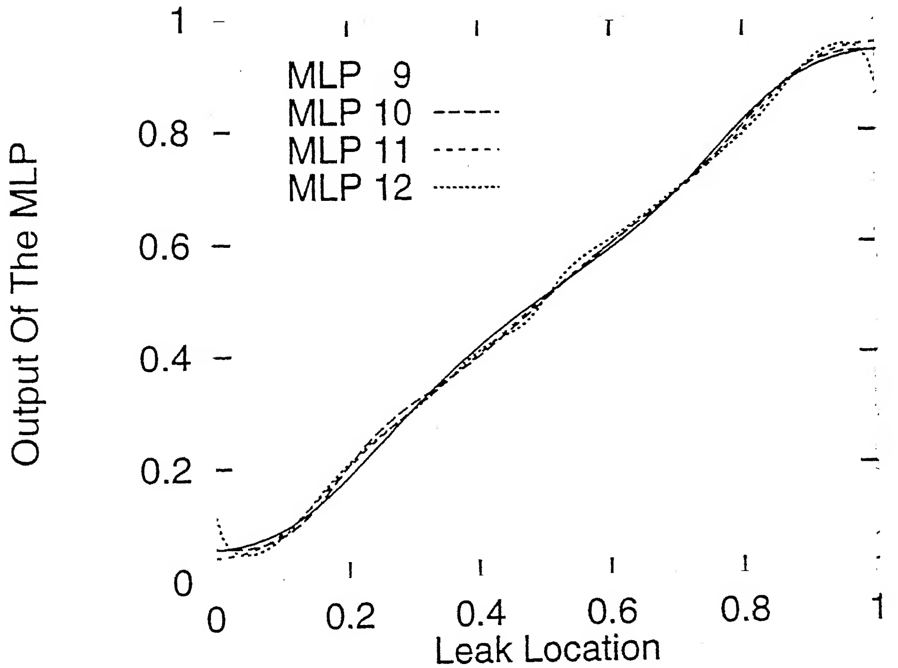

Figure 3.9: Results Of Testing For MLP 9, MLP 10, MLP 11 and MLP 12.

Leak Location	Outout Of MLP	Absolute Error	Ep
0.050000	0.054107	0.004107	8.213997
0.100000	0.080401	0.019599	19.598997
0.150000	0.132239	0.017761	11.840671
0.200000	0.203890	0.003890	1.944996
0.250000	0.266021	0.016021	6.408405
0.300000	0.307318	0.007318	2.439330
0.350000	0.344070	0.005930	1.694288
0.400000	0.391278	0.008722	2.180502
0.450000	0.449074	0.000926	0.205775
0.500000	0.505655	0.005655	1.130998
0.550000	0.555050	0.005050	0.918182
0.600000	0.603246	0.003246	0.540992
0.650000	0.646113	0.003887	0.598000
0.700000	0.688296	0.011704	1.671996
0.750000	0.742270	0.007730	1.030668
0.800000	0.807727	0.007727	0.965871
0.850000	0.869340	0.019340	2.275291
0.900000	0.914998	0.014998	1.666447
0.950000	0.942647	0.007353	0.774001
1.000000	0.953291	0.046709	4.670900

Table 3.17: Sample Results Of Testing For MLP 13.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.059202	0.009202	18.404005
0.100000	0.084095	0.015905	15.905000
0.150000	0.122369	0.027631	18.420671
0.200000	0.185117	0.014883	7.441498
0.250000	0.260754	0.010754	4.301596
0.300000	0.319122	0.019122	6.373991
0.350000	0.355542	0.005542	1.583431
0.400000	0.395441	0.004559	1.139753
0.450000	0.447269	0.002731	0.606888
0.500000	0.501572	0.001572	0.314403
0.550000	0.552924	0.002924	0.531630
0.600000	0.601738	0.001738	0.289659
0.650000	0.644378	0.005622	0.864918
0.700000	0.688176	0.011824	1.689145
0.750000	0.743965	0.006035	0.804663
0.800000	0.809796	0.009796	1.224495
0.850000	0.872047	0.022047	2.593763
0.900000	0.916249	0.016249	1.805445
0.950000	0.938718	0.011282	1.187575
1.000000	0.945184	0.054816	5.481601

Table 3.18: Sample Results Of Testing For MLP 14.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.059430	0.009430	18.859997
0.100000	0.073999	0.026001	26.000999
0.150000	0.124201	0.025799	17.199337
0.200000	0.202079	0.002079	1.039497
0.250000	0.264328	0.014328	5.731201
0.300000	0.307220	0.007220	2.406667
0.350000	0.344975	0.005025	1.435714
0.400000	0.393393	0.006607	1.651749
0.450000	0.460690	0.010690	2.375556
0.500000	0.517546	0.017546	3.509200
0.550000	0.537315	0.012685	2.306364
0.600000	0.586118	0.013882	2.313673
0.650000	0.652338	0.002338	0.359700
0.700000	0.706671	0.006671	0.953002
0.750000	0.751836	0.001836	0.244800
0.800000	0.801771	0.001771	0.221372
0.850000	0.866439	0.016439	1.933995
0.900000	0.916741	0.016741	1.860115
0.950000	0.936709	0.013291	1.399053
1.000000	0.940402	0.059598	5.959803

Table 3.19: sample Results Of Testing For MLP 15.

Leak Location	Output Of MLP	Absolute Error	Ep
0.050000	0.059101	0.009101	18.202003
0.100000	0.083764	0.016236	16.236000
0.150000	0.126195	0.023805	15.870005
0.200000	0.190759	0.009241	4.620500
0.250000	0.257518	0.007518	3.007197
0.300000	0.310950	0.010950	3.650000
0.350000	0.354995	0.004995	1.427148
0.400000	0.397988	0.002012	0.503004
0.450000	0.442796	0.007204	1.600888
0.500000	0.494668	0.005332	1.066399
0.550000	0.555878	0.005878	1.068722
0.600000	0.614176	0.014176	2.362659
0.650000	0.656667	0.006667	1.025695
0.700000	0.687506	0.012494	1.784853
0.750000	0.729408	0.020592	2.745597
0.800000	0.802508	0.002508	0.313498
0.850000	0.876167	0.026167	3.078468
0.900000	0.919384	0.019384	2.153781
0.950000	0.938762	0.011238	1.182945
1.000000	0.942670	0.057330	5.733001

Table 3.20: Sample Results Of Testing For MLP 16.

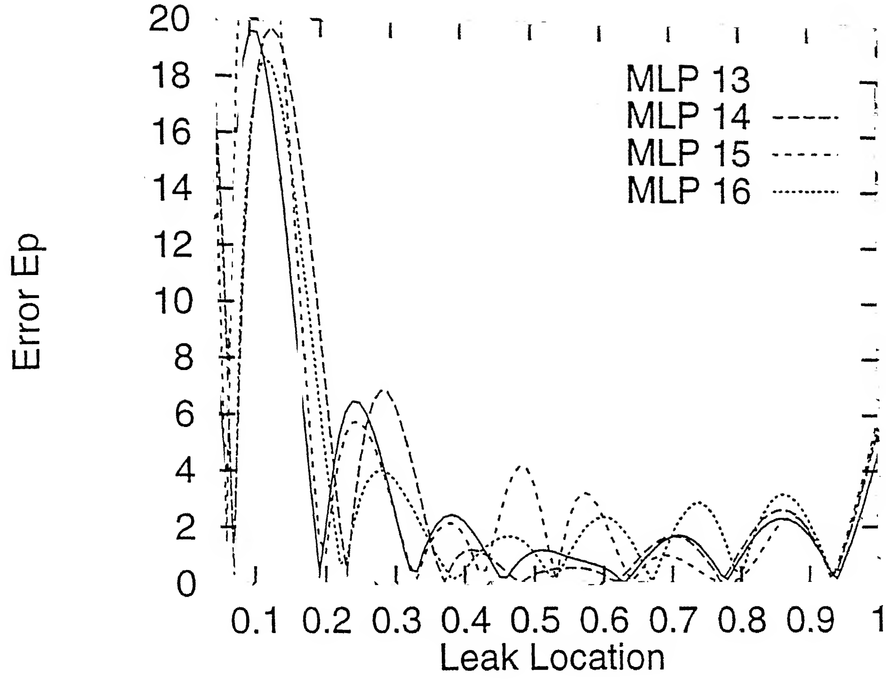

Figure 3.10: Plot Of Error (Ep) In Leak Localization For Different Leak Locations.

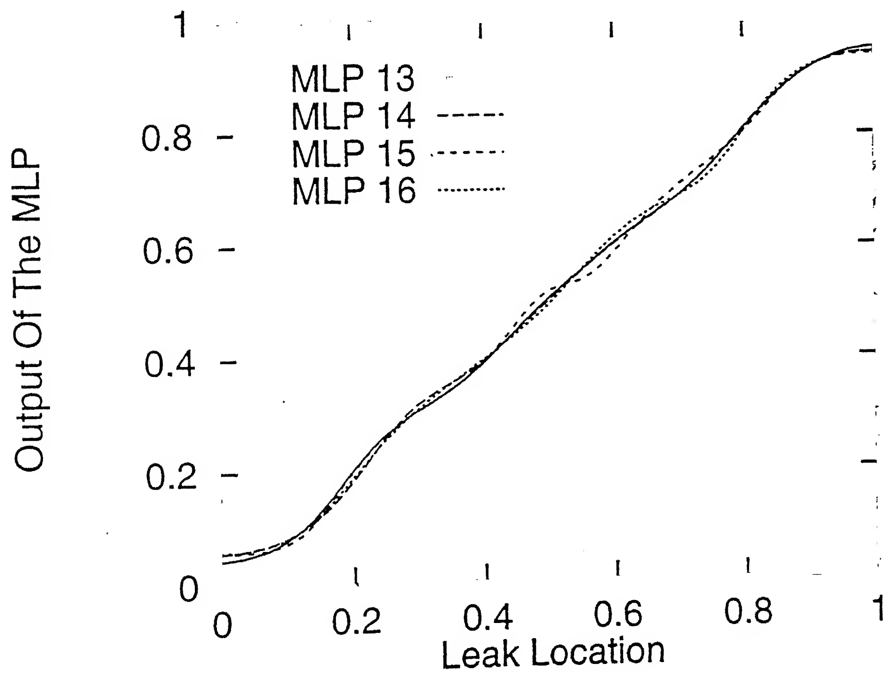

Figure 3.11: Results Of Testing For MLP 13, MLP 14, MLP 15 and MLP 16.

Steps of the **algorithm** are as follows

1. Set i = 1 and start with guess value Y_1.
2. Find search direction $S_i = -\nabla\mathcal{E}_i$ where $\nabla\mathcal{E}_i$ is the gradient of function $\mathcal{E}$ at the iteration i, i.e. for Signal 4 type of model,

$$\nabla\mathcal{E}_i = \begin{pmatrix} \partial\mathcal{E}/\partial\alpha_{1i} \\ \partial\mathcal{E}/\partial\alpha_{2i} \\ \partial\mathcal{E}/\partial\omega_{1i} \\ \partial\mathcal{E}/\partial\omega_{2i} \\ \partial\mathcal{E}/\partial c_{1i}c_{2i} \end{pmatrix}$$

3. Find λ_i^* to minimize $\mathcal{E}(Y_i + \lambda_i S_i)$.
4. Set $Y_{i+1} = Y_i + \lambda_i^* S_i$.
5. If Y_{i+1} is not optimum i.e. $\mathcal{E}_{i+1} \geq \epsilon$ go to step 2.
6. $Y_{optimal} = Y_{i+1}$.
7. Stop.

3.6.2 Results With The Real Data

From the Kandla-Bhatinda pipeline of IOCL some real data regarding the leak localization was obtained. It consisted of correlation patterns for five leak locations. One of the data was used for the purpose of optimization. Out of the four models, Signal 4 type of model was found to give better fit to the empirical values. The optimized value of error function $\mathcal{E}$ was obtained at values of parameters given in Table 3.21. Using these values in the model, $R_{P_AP_B}(\beta)$ patterns were generated for 101 different leak locations. The MLP was trained using these patterns. Then the MLP was tested for the reamining four real data's. The results of testing are shown in the Table 3.22. From the Table 3.22 it is clear that the strategy suggested in this section yields acceptable results. The percentage error Ep is around 5%. And the

α_1	α_2	ω_1	ω_2	c_1c_2
0.107	0.135	207	493	79

Table 3.21: Optimized parameters of Signal 4 type of model.

Leak Location	Output Of MLP	Absolute Error	Ep
0.270000	0.287596	0.017596	6.517037
0.430000	0.450957	0.020957	4.873209
0.670000	0.657204	0.012796	1.909857
0.720000	0.701514	0.018486	2.567500

Table 3.22: Results for real data using MLP.

absolute error is also small.

Although the method of using correlation patterns for the leak detection gives good results for single leak case, some assumptions have been made in this method which call for a different approach of solving the same problem. The MLP was trained to identify correlation patterns generated by a model, based on the assumption that the pressure wave generated by the leak propagates in both left and right directions. For multiple leak case the above assumption causes problems . And hence for multiple leak case as well as for complex networks of pipelines a totally different approach is necessary.

Chapter 4

Alternative Method: Use Of TDNN

In the solution of leak localization problem using the correlation method, the MLP was trained to differentiate between correlation patterns associated with different leak locations. Another approach of solving the same problem is feeding the pressure signals to an ANN which can extract the spatial as well as temporal variations in the pressure signals.

4.1 Introduction

The alternative method of feeding pressure signals directly to an ANN, which can extract the spatial as well as temporal features of the signal needs a different type of architecture called TDNN[11]. The BP algorithm has established itself as the most popular method for the design of ANNs. However a major limitation of the standard BP algorithm is that it can only learn an input output mapping that is static. Consequently the MLP so trained has a static structure that maps an input vector X onto an output vector Y.

Time is important in many of the cognitive tasks encountered in practice, such as vision and speech. How to extend the design of the MLP so that it assumes a time varying form and therefore will be able to deal with time varying signals is a

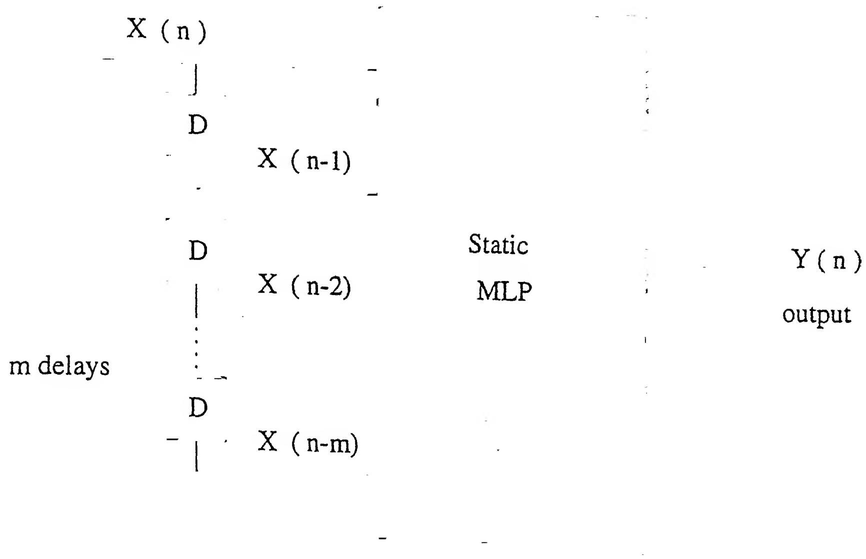

Figure 4.1: Basic Architecture Of TDNN.

major question. TDNN is a multilayer feedforward network with architecture as per Fig. 4.1.

4.2 Scheme Of Using TDNN

Figure 4.2 shows pipeline section for the present scheme. Let the section of pipeline under consideration has length $\nabla x(n - 1)$, n pressure sensors being located at distance ∇x from the previous sensor.

Let the n pressure sensors be denoted by P_1, P_2 ..., P_n

let $P_k(l)$ denote pressure at sensor k at instant l.

where k = 1, 2, 3, ... n

and l = 0, 1, 2, ... M

Let L_1, L_2 ... L_q denote leaks at distances d_1, d_2 ... d_q respectively from P_1. Figure 4.3 shows the architecture of the TDNN used for the present approach. The TDNN used in the discussion consists of $n(m+1)$ inputs , two hidden layers and q outputs at the output layer. Out of the total $n(m+1)$ inputs, n inputs are the instantaneous values of the pressures at n different locations. While the remaining nm inputs are the delayed pressure values at previous m instances at n different

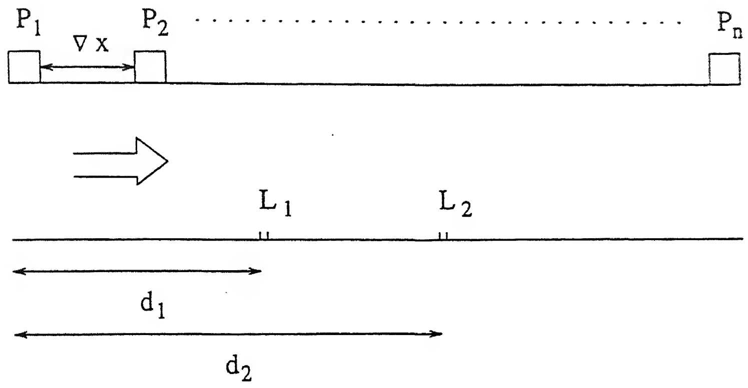

Figure 4.2: Sensor Locations.

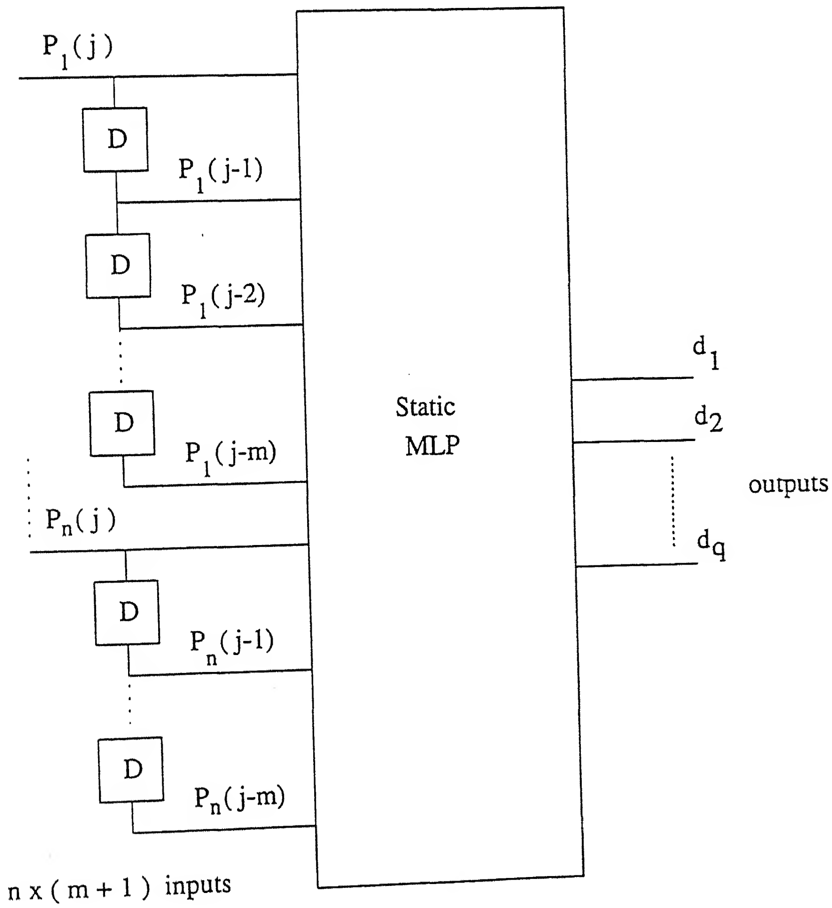

Figure 4.3: TDNN Architecture.

locations. Thus the input to the TDNN contains information regarding both the spatial as well as the temporal nature of the pressure signal. The TDNN is made to learn this nature of the pressure signal corresponding to particular leak location(s), in the training phase. Whether the TDNN has learned to extract the information regarding the location(s) of leak(s) from the inputs is tested during the testing phase.

4.2.1 Data Set Generation

Pressure signals used in this method are as follows

For single leak case

$$P_k(l) = e^{-\alpha_1 t}\, sin\omega_1 (t - \frac{|d_1 - \nabla x(k-1)|}{v}) + e^{-\alpha_2 t}\, sin\omega_2 (t - \frac{|d_1 - \nabla x(k-1)|}{v}) + a_k$$

where a_k is the steady state component of pressure signal.

For double leak case

$$P_k(l) = e^{-\alpha_1 t}\, sin\omega_1 (t - \frac{|d_1^* - \nabla x(k-1)|}{v}) + e^{-\alpha_2 t}\, sin\omega_2 (t - \frac{|d_2^* - \nabla x(k-1)|}{v}) + a_k$$

where a_k is the steady state component of pressure signal,

$d_1^* = |d_1 + d_2|/2$

and $d_2^* = |d_1 - d_2|/2$.

4.2.2 Implementation

Data sets were generated using above expressions. The different parameters used in this method were as follows. Number of sensors (n) was 5. Number of delays (m) was 4. There were 25 inputs and 1 output (for single leak case)/ 2 outputs (for double leak case). There were 2 hidden layers, with 15 neurons in the first hidden layer and 9 neurons in the second hidden layer. Twentyfive patterns consisting of 25 inputs and 1(2) output(s) for single (double) leak case were generated for 10 different time instances. Fifteen patterns were used for training while remaining

α_1	α_2	ω_1	ω_2	a_1	a_2	a_3	a_4	a_5
.1	.14	200	440	10	9.5	9	8.5	8

Table 4.1: Parameters For The TDNN.

were used for testing. Table 4.1 gives values of parameters used for this method.

4.3 Training Of The TDNN

A typical training pair consists of leak location(s) as the output(s) for single(double) leak case and $n(m+1)$ i.e. 25 inputs, $P_1(j)$, $\ldots$ $P_1(j-4)$ $\ldots$ $P_5(j-4)$. Since the pressure signal varies as a function of time, such M (presently 10) training pairs constitute complete information about the leak location(s).

Data sets generated as discussed in previous section were used during the training phase. At the start of training, all weights were randomly generated. At a time, a training pair consisting of 25 inputs and 1(2) output(s) was fed to the TDNN. The weights were updated after calculating error for the training pair. Such 10 pairs corresponding to particular location(s) of leak were fed to the TDNN, which corresponded to the temporal variation of the signal for the same leak location(s). Two different TDNNs were trained, one for single leak case and other for double leak case. The training was stopped when the error E as defined in *Chapter 3* was less than 2%.

4.4 Testing Of The TDNN

4.4.1 Single Leak Case

In the testing phase, ten patterns were tested. The testing patterns include patterns for different leaks, as well as for same leak locations but at different time intervals. Table 4.2 shows the results of testing.

Pattern	j	d_1	Output Of TDNN	Absolute Error	Ep
1	5	.12	.096754	0.023246	19.371666
2	8	.16	.132914	0.027088	16.930000
3	6	.36	.374531	0.014531	4.036388
4	9	.36	.362481	0.002461	0.683607
5	7	.44	.469210	0.003608	0.819997
6	9	.64	.636215	0.003785	0.591408
7	5	.64	.621343	0.018657	2.915156
8	7	.72	.708849	0.011151	1.548752
9	6	.88	.901276	0.021276	2.417727
10	7	.96	.941080	0.018920	1.970834

Table 4.2: Results For Single Leak Case Using TDNN.

4.4.2 Two Leak Case

The TDNN trained during the training phase was tested for ten patterns. The results of the testing are as shown in Table 4.3.

4.4.3 Results

Tables 4.2 and 4.3 show the performance of the TDNN for single and double leak case. A typical testing pattern corresponds to pressure values at five different locations for the present as well as four previous time instances, corresponding to particular leak position(s) . Thus the pressure signals fed to the TDNN in testing phase depend on the time instant (j) chosen. For the single leak case (Table 4.2) testing patterns 3 and 4 corresponded to same leak location (.36) at different time instances. Also patterns 6 and 7 corresponded to same leak location. Similarly for the double leak case also testing patterns 3 & 4 and 6 & 7 corresponded to same set of leak locations at different time instances. From Tables 4.2 and 4.3 it is clear that the properly trained TDNN can detect leak locations in single leak as well as

Testing pattern	j	d_1	Output 1	d_2	Output 2
1	7	.2	.18	.4	.44
2	8	.4	.43	.2	.18
3	6	.4	.42	.6	.63
4	8	.4	.41	.6	.62
5	6	.6	.64	.2	.24
6	5	.2	.19	.8	.76
7	9	.2	.21	.8	.79
8	6	.8	.77	.6	.63
9	7	1	.98	.8	.76
10	9	.6	.53	.8	.81

Table 4.3: Results For Double Leak Case Using The TDNN.

d_1	Output Of TDNN	Ep For TDNN	Output Of MLP	Ep For MLP
.12	.096754	19.371666	0.088337	26.385833
.16	.132914	16.930000	0.139348	12.907500
.36	.362481	0.683607	0.353354	1.846111
.36	.362481	0.683607	0.353354	1.846111
.44	.469210	0.819997	0.446247	1.419772
.64	.636215	0.591408	0.639731	0.042031
.64	.636215	0.591408	0.639731	0.042031
.72	.708849	1.548752	0.725217	0.724583
.88	.901276	2.417727	0.900736	2.356363
.96	.941080	1.970834	0.938437	2.246145

Table 4.4: Results For Single Leak Case Using TDNN and MLP : A Comparison.

double leak case with acceptable accuracy.

4.4.4 Comparison Of TDNN and MLP Approaches For Single Leak Case

Table 4.4 shows the results of TDNN and MLP trained and tested on data sets produced using the same model. The performance of both the methods is not much different. Training of TDNN is more complex than that for MLP. The TDNN was trained for 40000 training cycles, while the MLP which gave similar performance took 20000 training cycles.

Chapter 5

Conclusion

The thesis discussed two possible methods of tackling the leak localization problem using ANNs. Performance of both the classifiers was tested largely on simulated data. Both classifiers i.e. MLP and TDNN performed well on the test data.

One of the interesting idea used in the thesis was to use the MLP for classifying correlation patterns of pressure signals, associated with different leak locations. The MLP was tested for both real as well as simulated data. The performance of this method was found to be satisfactory for single leak case.

The other approach of using an ANN architecture which can extract the spatial as well as temporal features in pressure signals, proved to be successful for single leak as well as double leak case. The complexity of the TDNN obviously increases burden on the training.

The use of TDNN approach, for multiple leak cases as well as for complex pipeline networks will be an interesting direction for future to come.

Bibliography

[1] Waterbury Robert C. , "Noise correlator spots pipe leaks," INTECH, pp 41-42, September 1992.

[2] Lux Edward and Harold Bob, "Telematry system monitors pipline," INTECH, pp 42-43, May 1993.

[3] Klein R. L. , *Pipelines: Design, construction and operation*, New York Construction Press, 1984.

[4] Lange F. H. , *Correlation Techniques Foundations and applications of correlation analysis in modern communications, measurement and control*, London ILIFFE Books Ltd., 1967.

[5] Burdic William S., *Underwater acoustic system analysis*, Prentice-Hall Signal Processing Series, 1984.

[6] Papoulis Athansios, *Probability, random variables and stochastic processes*, Third Edition, Singapore McGraw Hill Book Co., 1991

[7] Finkelstein L. and Gratton K.T.V. , *Concise Encyclopedia of Measurement and Instrumentation*, Pergamon Press Ltd., 1994.

[8] Lippman Richard P. , "Introduction to Computing with Neural Nets," IEEE Signal Processing Magazine, pp 4-22, April 1987.

[9] Bishop Chris M. , " Multiphase Flow Monitoring In Oil Pipelines," Applications of Neural Networks, Kluwer Academic Publishers pp 133-155, 1995.

[10] Kosko Bart , *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, New Delhi: Prentice Hall India, 1992.

[11] Haykin Simon , *Neural Networks A Comprehensive Foundation*, Macmillan College Publishing Company Inc. 1994.

[12] Press William , Teukolsky Saul , Vellering William and Flannery Brain , *Numerical Recipes In C*, Second Edition, Cambridge University Press, 1992.

[13] Rao S S , *Optimization theory and applications*, Second Edition, Wiley Eastern Limited, 1984.

[14] Isermann Rolf, " Process Fault Detection Based on Modeling and Estimation Methods - A Survey, " Automatica, vol.20 No.4 pp 387-404, 1984.

EE-1886-M-KUL-NEU.